EE422C Project 3 (Word Ladder) Test Plan

Manuel Gomez mlg3454
Don Ton dt22776
Fall 2016

We tested our program one method at a time as we worked through the project. Once we felt a method was done we would set up our main to debug that specific method. Once a bug was found, we implemented the necessary changes and moved onto the next part of our program. Junit testing was not used until we felt we had completed the program. Our primary method of debugging was to use debugging mode and enter in different test cases into the console. Junit was used towards the end of the project to debug edge cases. Overall, the use of the debugger to solve obvious bugs was helpful while working through the project, and Junit test was helpful in identifying problems we initially did not think of.

BFS:

1.
    a. ShortBFS
    b. This test covers if the BFS search method can quickly find a short word ladder
    c. Ran initialize()
    d. Ladder between “hello” and “cells” that should have less than 4 rungs.
    e. No stack overflow, ladder has no duplicates, and ladder less than 4 rungs.
    f. No comments
2.
    a. NoLadder
    b. This test covers using the BFS search where no ladder exists
    c. Ran initialize()
    d. No ladder exists
    e. The test passes if it shows that no ladder exists
    f. No comments
3.
    a. OneDiff
    b. This test covers the BFS search where the two inputs differ by one letter
    c. Ran inialize()
    d. No rungs on ladder, only start and end word
    e. The test passes if only the start and end words are in the output
    f. No comments
4.
    a. Quit
    b. The test covers the /quit command
    c. Ran initialize()
    d. No output
    e. The test passes if the program terminated with no output

5.
   a. Spaces
   b. The test covers the BFS search method where there are extra spaces in the user input
   c. Ran initialize()
   d. The output should match the output of the first test
   e. The test passes if the correct word ladder is displayed when multiple spaces separate the inputs
   f. Extra spaces between start and end word

DFS:

1.
   a. LargeLadder
   b. Test ensure that a large ladder will not have a stack overflow
   c. Ran initialize()
   d. The output should be a ladder with over 3000 rungs
   e. Passes if ladder is produced with no repeats and no stack overflow in under 30s
   f. No comments
2.
   a. Spaces
   b. The test covers the DFS search method where there are extra spaces in the user input
   c. Ran initialize()
   d. The output should be the expected word ladder
   e. The test passes if the correct word ladder is displayed when multiple spaces separate the inputs
   f. Extra spaces between start and end word
3.
   a. Quit
   b. The test covers the /quit command
   c. Ran initialize()
   d. No output
   e. The test passes if the program terminated with no output
   f. No comments
4.
   a. NoLadder
   b. This test covers using the DFS search where no ladder exists
   c. Ran initialize()
   d. No ladder exists
   e. The test passes if it shows that no ladder exists
   f. No comments
5.
   a. Quit with spaces
   b. This test covered whether extra spaces with the /quit command will still terminate the program
   c. Ran initialize()

d. No output
e. The test passes if the program terminates with no output
f. Spaces placed before and after /quit